Design of an Integrated Web Services Brokering System

Frederick Petry, Naval Research Laboratory, Stennis Space Center, USA
Roy Ladner, Naval Meteorology and Oceanography Command, Stennis Space Center, USA
Kalyan Moy Gupta, Knexus Research, USA
Philip Moore, Knexus Research, USA
David W. Aha, Naval Research Laboratory, USA
Bruce Lin, Naval Research Laboratory, Stennis Space Center, USA
Richard Sween, Naval Research Laboratory, Stennis Space Center, USA

ABSTRACT

This paper describes an Integrated Web Services Brokering System (IWB) to support the automated discovery and application integration of Web Services. In contrast to more static broker approaches that deal with specific data servers, our approach creates a dynamic knowledge base from Web Service interface specifications. This assists with brokering of requests to multiple data providers even when those providers have not implemented a community standard interface or have implemented different versions of a community standard interface. A specific context we illustrate here is the domain of meteorological and oceanographic (MetOc) Web Services. Our approach includes the use of specific domain ontologies and has evaluated the use of case-based classification in the IWB to support automated Web Services discovery. It was also demonstrated that the mediation approach could be extended to OGC Web Coverage Services.

Keywords: Case-Based Classifier; Mediation; MetOc; OGC; Ontology; Web Services

INTRODUCTION

Web Services are becoming the technology used to share data in many domains. Web Services technologies provide access to discoverable, self-describing services that conform to common standards. Thus, this paradigm holds the promise of an automated capability to obtain and integrate data. While desirable, access and retrieval of data from heterogeneous sources in a distributed system such as the Internet pose many difficulties and require efficient means of discovery, mediation and transformation of requests and responses. Differences in schema and terminology prevent simple querying and retrieval of data. These functions require processes that enable identification of appropriate services, selection of a service provider of requested data, transformation of requests/responses, and invocation of the service interface. Service availability must also be resolved. There have been a variety of approaches developed for these functions, but

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2009		**00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Design of an Integrated Web Services Brokering System	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Knexus Research Corp,9120 Beachway Lane,Springfield,VA,22153**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
to appear in Journal of Information Technology and Web Engineering

14. ABSTRACT
This paper describes an Integrated Web Services Brokering System (IWB) to support the automated discovery and application integration of Web Services. In contrast to more static broker approaches that deal with specific data servers, our approach creates a dynamic knowledge base from Web Service interface specifications. This assists with brokering of requests to multiple data providers even when those providers have not implemented a community standard interface or have implemented different versions of a community standard interface. A specific context we illustrate here is the domain of meteorological and oceanographic (MetOc) Web Services. Our approach includes the use of specific domain ontologies and has evaluated the use of case-based classification in the IWB to support automated Web Services discovery. It was also demonstrated that the mediation approach could be extended to OGC Web Coverage Services.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**22**	

primarily independently of each other and not fully automated, that is, often requiring human intervention.

In this paper we describe the design of an integrated end-to-end brokering system that performs automated discovery, automated mediation and automated transformation of Web Services requests and responses. In contrast to more static approaches that deal with pre-selected data servers, our approach creates a dynamic knowledge base from Web Service interface specifications that are discovered on the fly. The dynamic knowledge base assists with mediating requests to data providers that have ad-hoc interfaces or differing versions of a community accepted interface.

Our design incorporates ontologies into the development of an Integrated Web Services Broker (IWB). This approach contrasts with developments that assume that shared ontologies have been adopted or published in order to support service discovery and integration. In addition to the use of ontologies we have evaluated classifier technology for the subtask of Web Services discovery. It has been noted that classifiers generalize well in sparse data, which is a characteristic of our Web Services application domain. Our use of classifiers in this manner does not require a formal domain definition nor does it require data providers to deploy any additional specialized ontological descriptions of their Web service.

There are general characteristics that should be found in any environment in which an automated system will operate. First, the domain must be one in which human intervention is neither required nor desirable. Since we are considering a Web Services context, data providers must have adopted a text-based, structured Web Services interface. This interface should subscribe to Web Services standards of self-description. While the structural content of each Web Services interface may differ significantly, the domain should be one in which key terminology that may be found in any interface has common conceptual content and is well understood and bounded. In this operating environment, it is desirable to isolate potential data sources in advance as opposed to attempting to discover service availability and capability on demand.

These characteristics are broadly applicable and encompass many typical application areas, and in this paper we describe the design and development of the IWB relative to these characteristics. The steps and issues we will be describing are basically applicable to any Web Services brokering system. Here, we illustrate the design specifically for the application context in which we are developing this system, that is, meteorological and oceanographic (MetOc) forms of data.

BACKGROUND

Web Services

Web Services provide data and services to users and applications over the Internet through a consistent set of standards and protocols. The most commonly used standards and protocols include, but are not necessarily limited to, the Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), the Web Services Definition Language (WSDL) and Universal Discovery Description and Integration (UDDI) (Cerami, 2002).

XML is a language used to define data in a platform and programming language independent manner. XML has become one of the widely used standards in interoperable

exchange of data on the Internet but does not define the semantics of the data it describes. Instead, the semantics of an XML document are defined by the applications that process them. XML Schemas define the structure or building blocks of an XML document. Some of these structures include the elements and attributes, the hierarchy and number of occurrences of elements, and data types, among others (Dick, 2000).

WSDL allows the creation of XML documents that define the "contract" for a Web service. The "contract" details the acceptable requests that will be honored by the Web service and the types of responses that will be generated. The "contract" also defines the XML messaging mechanism of the service. The messaging mechanism, for example, may be specified as SOAP. A Web service describes its interface with a WSDL file and may be registered in a UDDI type registry. Interfaces defined in XML often identify SOAP as the required XML messaging protocol. SOAP allows for the exchange of information between computers regardless of platform or language.

A registry provides a way for data providers to advertise their Web Services and for consumers to find data providers and desired services (Figure 1). It is, of course, not necessary to register a Web service. However, that would be similar to a business not listing its telephone number in a telephone directory. Not having a listing would make it more difficult for consumers to discover and utilize a Web service. This advertisement of Web Services may or may not be desirable for net-centric operations in many application communities such as those found in many military operations (Ladner & Petry, 2005).

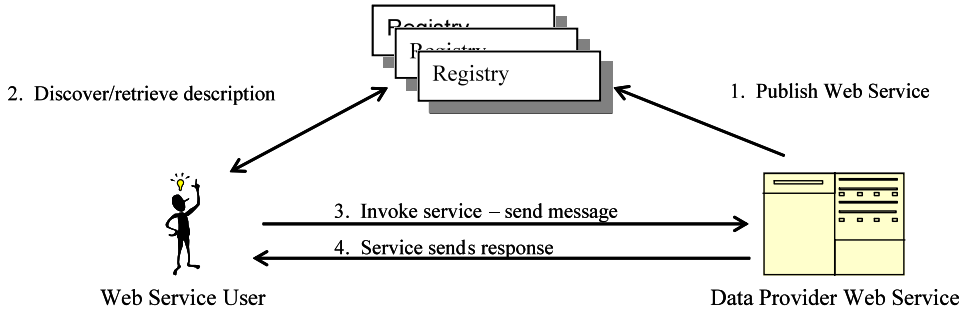


Figure 1. Illustrated use of Web Services

There are applications that provide services on the Web without using all components of the Web Services stack. These Web-based services employ diverse methods for discovery, description, messaging and transport. Within these Web-based services adherence to standards and protocols vary. There has been some work in applying the soft computing techniques of fuzzy set theory particularly for discovery (Chao Younas, Lo, & Tan, 2005; Fenza, Loia, & Senatore, 2007, 2008).

Since we are interested in querying MetOc data some of which is available from DoD, the Joint MetOc Broker Language (JMBL) was a basis of the query format for the IWB. JMBL (JMBL, 2009) defines a syntax that allows standardized request and response structures for MetOc data queries. It was developed with input from joint forces including Army, Navy, Air Force, and so forth. A goal of JMBL was to define one Web

Service based on jointly defined XML Schemas that would serve all types of MetOc data requests. Each agency would implement this jointly defined Web Service and would therefore have interoperable implementations of the same Web Service.

The JMBL Web Service is defined by one WSDL file and several XML Schemas. These Schemas define the structure of requests that the JMBL Web Service will accept and the structure of responses that the JMBL Web Service will provide. The request and response Schemas include several other Schemas, which define global data types and structures. Figure 2 shows this conceptual organization. As shown in Figure 2, several of the global Schemas are included in other global Schemas. Schemas in Figure 2 are represented by "XSD".

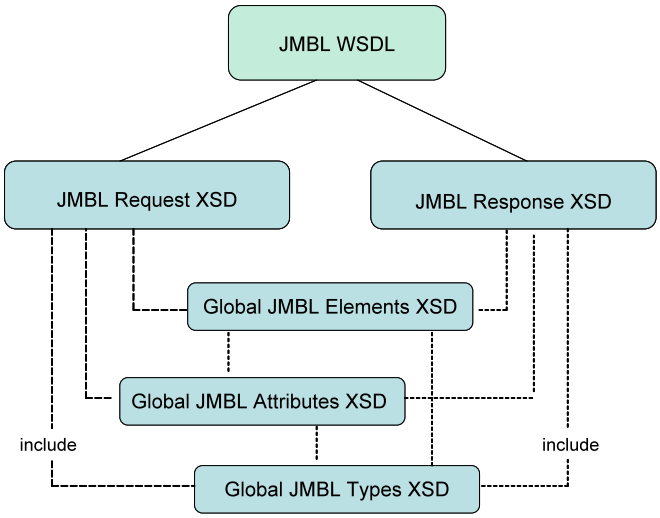


Figure 2. Conceptual view of JMBL WSDL and schemas

Ontologies

The word "ontology" originated in philosophy and means a theory of the nature of existence or the science of being. In particular, it describes what types of things can exist. As applied in computer systems some typical definitions could be the following:

- An ontology is a formal shared conceptualization of a particular domain of interest (Stojanovic et al., 2004).
- An ontology is an explicit specification of an abstract, simplified view of a world we desire to represent (Holsapple & Joshi, 2002).

An ontology defines common words and concepts used in the description of an area of knowledge to share domain information for applications. An ontology is usually structured by including classes representing concepts, instances of classes and the relationships among them. Also included are their properties and functions or processes related to them.

There are two approaches that can be taken to ontology development, top-down and bottom-up approaches. Many ontologies have been constructed by top-down iterative development. This falls in the same approach as the design of taxonomies and

classifications by a specialization process. Bottom-up design techniques typically begin by specifying the list of primitive concepts and construction rules that define more complex concepts, that is, a generalization process. In practice it is often a mix of these two approaches that are actually utilized. Building ontologies is typically difficult, time-consuming and expensive. This is especially so, if the goal is to construct an ontology that is rich and powerful enough to perform automated inferencing. Construction of such an ontology requires careful attention to detail and a strong ability to organize information meaningfully. It is often stated that ontology development is a craft rather than a science (van der Vet & Mars, 1998).

Ontological frameworks for describing the semantics of data include such developments as the Resource Description Framework (RDF) and Web Ontology Language (OWL). RDF provides a flexible representation of information and a reliable means of supporting machine reasoning (Powers, 2003). OWL permits users to more fully describe the meanings of terms found in Web documents and to represent the relationships among these terms (Lacy, 2005). Numerous methodologies for engineering and maintaining domain ontologies have been reported (Cristani & Cuel, 2004). There are also editors that assist with ontology development, such as the open source editor Protégé. A Protégé extension supports OWL ontologies (Knublauch, Fergerson, Noy, Musen, 2004). Even with these tools, ontology development remains a time- and skill-intensive activity.

Utilization of ontologies as metadata for various data sources on the semantic Web is of specific concern (Kim, 2002). Recent efforts to improve interoperability include Web Services technologies such as WSDL and XML Schemas. While these provide structured content, the limited nature of the semantic description hinders interoperability. Ontologies are often considered to be the basis of semantic meaning for these sorts of documents. OWL-S has been developed to extend OWL to supply the constructs for defining an ontology of services that is intended to support automated Web Services discovery, invocation, and composition. This is accomplished in the OWL-S ontology through classes that describe what the service does (service profile), how to ask for the service, what happens when the service is carried out (service grounding), and how the service can be accessed (service model) (W3C Member Submission, 2004).

IWB ARCHITECTURE

This section describes many of the processes and architectural features of an automated brokering system. We will illustrate this for the specific instance of our prototype application context, the MetOc Web Services domain. Meteorological and oceanographic Web Services provide actual and forecast weather information to a variety of government and commercial entities and the public. The information they provide can vary considerably depending on their intended audiences. Consequently, for an end-user, selecting and interacting with suitable Web Service(s) poses a significant challenge.

IWB Functionality

We are designing the IWB to automatically discover MetOc Web Services and then to dynamically translate data and methods across them. The IWB's Web Services search

and discovery function is illustrated in Figure 3. The IWB will search a variety of identified registries for MetOc Web Services using the search feature supplied by that registry. This then enables the IWB to locate candidate sources to which requests may be brokered. Based on the characteristics of the Web Services it discovers, the IWB builds a dynamic knowledge base to support mediation.

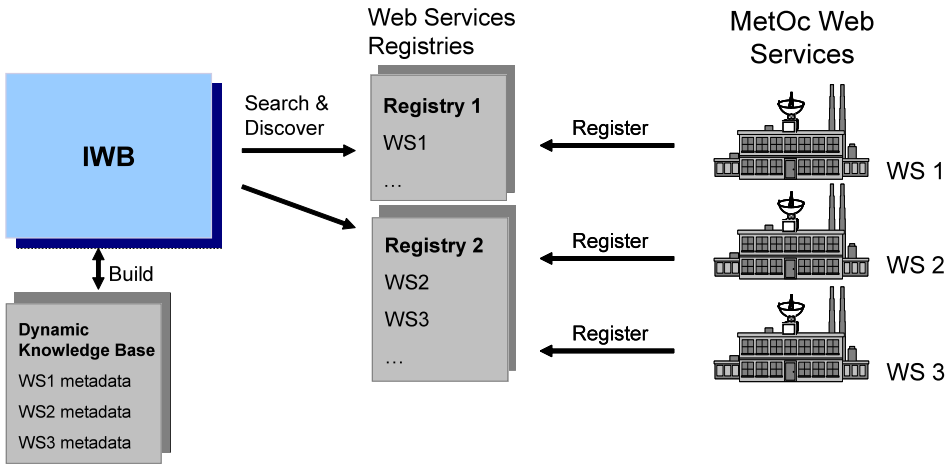


Figure 3. The IWB search and discovery function.

This knowledge base allows the IWB to automatically translate user requests to differing Web Service interface specifications. For example, this shall assist with brokering requests to multiple MetOc data providers whose services may have implemented a) a community standard interface, b) an interface that is not a community standard, or c) an evolving version of a community standard interface. This approach contrasts with approaches that use pre-programmed solutions for pre-selected data servers. The IWB's mediation function is depicted in Figure 4. The client request is then dynamically translated and mediated to Web Services with differing WSDLs/Schemas.

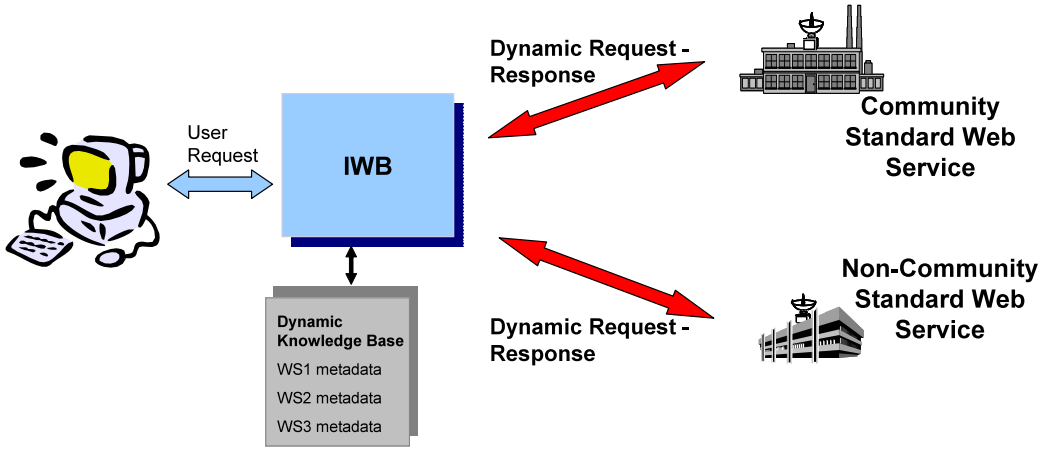


Figure 4. The IWB dynamic mediation function

IWB Processes

The high level processes at work in the IWB are Web Services discovery and mediation/transformation of user data requests. In this section we will describe the individual steps of each of these processes.

Web Services Discovery

To prepare for the Web service discovery process, the IWB first loads the functional ontology. This ontology allows IWB to interpret terms found in WSDLs and schemas, as necessary, in order to build the Dynamic Knowledge Base.

The actual search function of the IWB entails a capability to search specific registries. Our approach is to query these registries for Web Services whose name or description contains relevant keywords. For example, a name/description keyword list might be (metoc, ocean, atmosphere, temperature, etc.). The search will examine UDDI registries that may be applicable to this domain, as well as other known Web Services registries such as xmethods or Binding Point. Relevant WSDLs and corresponding schemas of identified Web Services are then downloaded.

The next step is the processing of the discovered WSDLs. This step involves the examination of each newly discovered WSDL and recording particular information about the Web service to enable mediation. The WSDL is decomposed into a symbol table of its contents so that available methods and their inputs can be easily identified. We use terms found in these methods and their inputs to identify those methods that are most likely to be MetOc data relevant. Following this is the creation of a blank XML message conforming to the required input of each of the identified methods. Finally, the structure of this XML message is mapped to ontology concepts. That is, for each term in the blank XML message, we determine which concept in the ontology it is related to. This permits the content of a client request to the IWB to be mapped to the target blank XML structure. Now it is possible for the IWB to add the newly found Web service to its Dynamic Knowledge Base (DKB). The DKB provides a quick means of identifying Web Services that provide specific data and data types, and it is updated every time the IWB identifies a new Web service or detects an update to a previously discovered Web service. The records comprising the DKB are built as follows. For each MetOc parameter supplied by the identified WSDL data retrieval method, the following is performed:

a) Retrieve the concept from the ontology.
b) Complete the blank XML template with the parameter name (“sal”, “depth” etc.). Associate the term used by the Web service with the concept from the ontology.
c) Create a Web service method record including the method name, xml message and XML to Ontology map as shown in Figure 5.
d) Record the Web service method record in the Dynamic Knowledge Base.

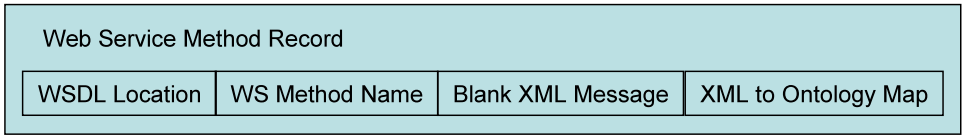
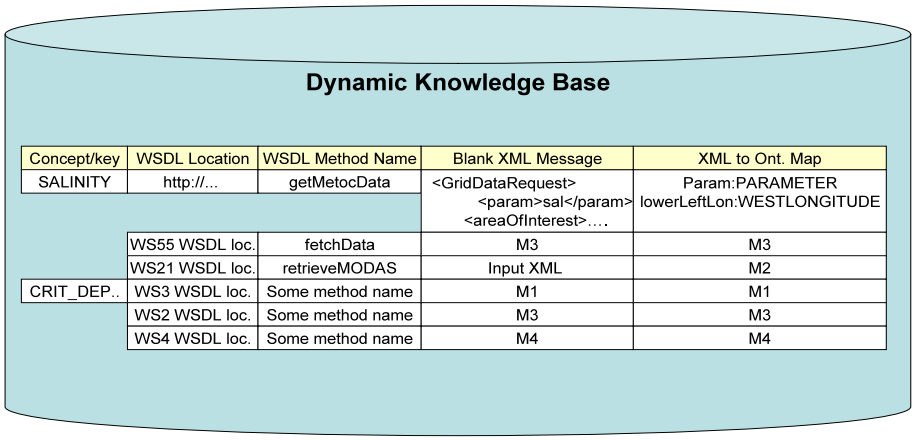


Figure 5. Web Service method record

Next we discuss a typical example of the indexing required for the Dynamic Knowledge Base as shown in Figure 6. The index key is the domain concept relevant to the parameters the Web service provides. These parameters are identified from terms found in the Web service's WSDL and schema. For example, a Web service, which contains oceanographic data such as "sea salinity" as an enumerated parameter, would be indexed by the concept "salinity."



Concept/key	WSDL Location	WSDL Method Name	Blank XML Message	XML to Ont. Map
SALINITY	http://...	getMetocData	<GridDataRequest> <param>sal</param> <areaOfInterest>….	Param:PARAMETER lowerLeftLon:WESTLONGITUDE
	WS55 WSDL loc.	fetchData	M3	M3
	WS21 WSDL loc.	retrieveMODAS	Input XML	M2
CRIT_DEP..	WS3 WSDL loc.	Some method name	M1	M1
	WS2 WSDL loc.	Some method name	M3	M3
	WS4 WSDL loc.	Some method name	M4	M4

Figure 6. IWB Dynamic Knowledge Base

Not shown in Figure 6 is the additional information necessary to mediate user requests, including each element/attribute that the schema identifies as mandatory, the SOAP Action & Service endpoint, the location for which data is provided and the type of MetOc data provided (such as grids, observational data, imagery, etc.).

Web Service Mediation

We now describe the second high-level process of the IWB; namely, the mediation of user requests for data. This step includes the transformation of user requests and Web Services responses. The steps involved are:

1. Receive an XML formatted user request for data.
2. Decompose the user request to identify those XML tags that have associated values.
3. Locate the tag that corresponds to a "parameter" synonym. This tag identifies the data request using the end-user's terminology.
4. Query the ontology for the concept corresponding to the term provided by the user.
5. Query the Dynamic Knowledge Base by this concept to obtain all Web Services that provide data related to the concept.
6. Transform the user's request to target Web service's request structure.

   Where the request must be brokered to multiple Web Services, there may be multiple transformations. This step utilizes the XML template recorded during the discovery process. This transformation is illustrated below. Figure 7a shows a request received by the IWB and the ensuing transformed request to be submitted to a Web service. Figure 7b further depicts the decomposed IWB

request and the indexed Web service request object. Transformation of the server response proceeds in a similar manner.

```
<GridRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <Parameter>salinity</Parameter>
        <aoi westLon="-90" southLat="10" eastLon="-80" northLat="20"/>
</GridRequest>
```

AMB sample request XML message

becomes

Web Service complete request XML message

```
<GridDataRequest xmlns="urn:nrl:metoc">
    <param>sal<param />
    <areaOfInterest>
        <westLongitude>-80 <westLongitude />
        <southLatitude>10 <southLatitude/>
        <easttLongitude>-70< easttLongitude />
        <northLatitude>20<northLatitude/>
    </areaOfInterest>
</GridDataRequest>
```

Figure 7a. Request Transformation

Decomposed User request

Concept	Request XML tag name	Request XML tag value	Attribute or element
PARAMETER	Parameter	salinity	element
WESTLONGITUDE	westLon	-80	attribute
SOUTHLATITUDE	southLat	10	attribute
EASTLONGITUDE.	eastLon	-70	attribute
NORTHLONGITUDE	northLat	20	attribute

becomes

XML to Ontology mapped request object

Concept	XML tag name	XML tag value	Attribute or element
PARAMETER	param	sal	element
WESTLONGITUDE	westLongitude		element
SOUTHLATITUDE	southLatitude		element
EASTLONGITUDE.	easttLongitude		element
NORTHLONGITUDE	northLat		element

Figure 7b. Request Decomposition example

IWB High Level Architecture

Here we will describe in some detail the architecture that integrates the processes described in the previous section. The functional components of the IWB are shown in Figure 8. The IWB can begin mediating user requests once its Mapper component has discovered Web Services and begun populating the Dynamic Knowledge Base. Specifically, the Mapper takes as *input* (1) discovered Web Services interface specifications and (2) the MetOc ontology. It uses this information to build the Dynamic Knowledge Base, and it also assigns a qualitative and quantitative confidence score to each service.

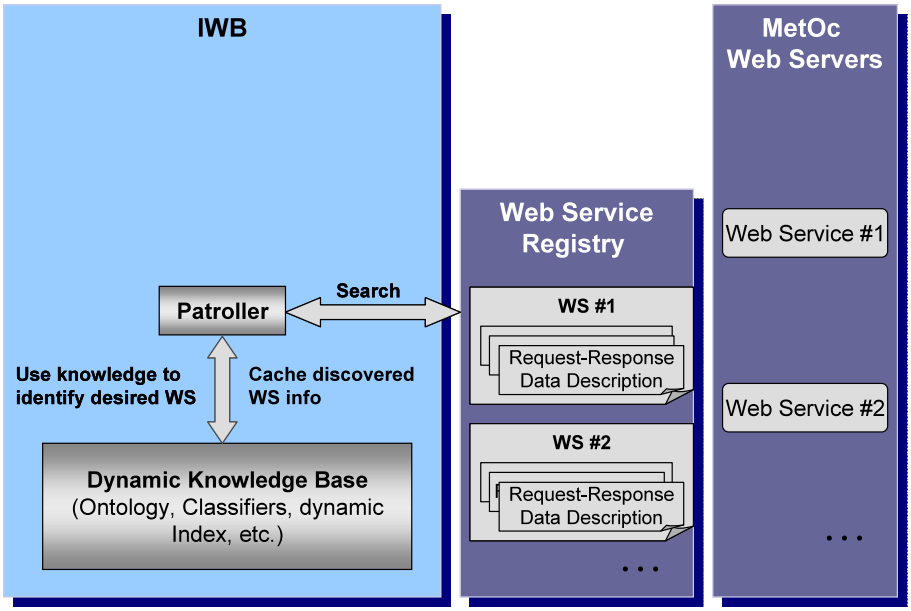


Figure 8a. IWB Architecture – Dynamic Discovery

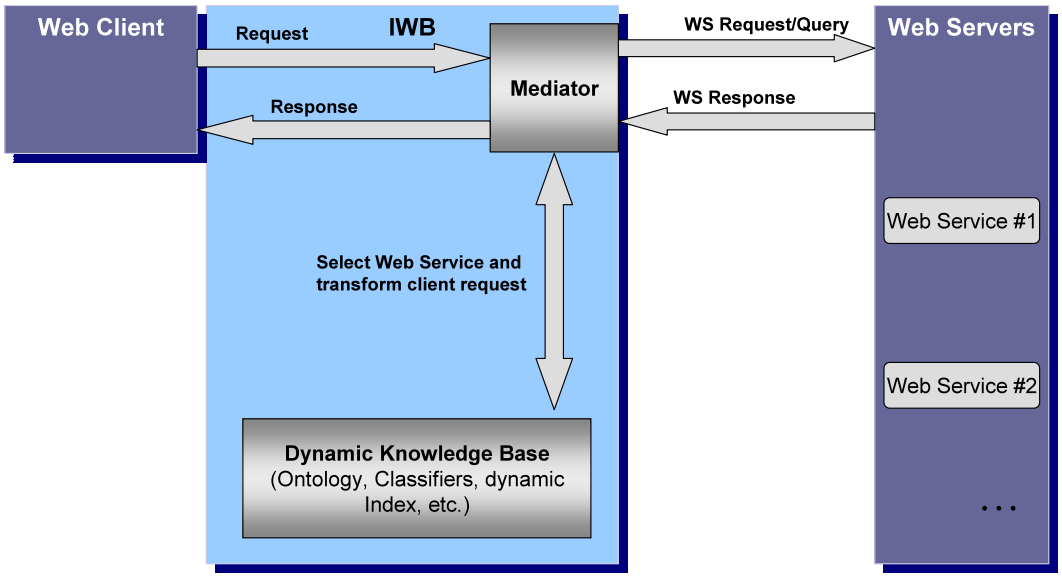


Figure 8b. IWB Architecture – Dynamic Mediation

After the IWB is initialized it is ready to process user requests to the appropriate Web service or multiple services. The Mediator is the component of the IWB that provides the necessary transforms for this to occur. Clients submit data requests to the Mediator in an IWB XML format. The Mediator uses the previously created mappings to translate the client request into a candidate Web service format specified in the Web Services Registry and submits the request to the Web service provider. As the recipient Web service sends the data response back to the Mediator, the Web service response is transformed by the Mediator to the end-user format and forwards it to the IWB's Client. This is the inverse of the request mapping process.

Partial Matching in IWB

As discussed, IWB performs two tasks: automated discovery and classification of Web services that produce MetOc data, and syntax-independent consumption of this data by clients utilizing an ontology of domain information for identification of MetOc services. For instance, the ontology captures the top-level concept of a MetOc "Parameter." An instance of this class, such as "Sea Temperature" may have synonyms: "SeaTemp" and "TempSea." As a new Web service is corralled by the IWB, its service description is broken into lexemes and matched to terms in the ontology. The ontology is manually constructed and maintained by domain experts, which results in a concise data model. However, small variations in a service description may thwart proper classification. For example, a service which offers a sea temperature parameter as "sTemp" may fail precise term matching, but there may be enough information to facilitate semi-automated ambiguity resolution. Another problem encountered while trying to index some Web services was the non-uniformity of labeling and describing Web services. For any given concept in the ontology, there could be many different synonyms that mean the same thing. Some services were labeled with terms that were similar to concepts in the ontology, but not exact matches. One example is the term "temperature." Using just the term, it is unclear whether the Web service provides air temperature, sea temperature, surface temperature, and so forth.

The IWB therefore employs a partial matching system to insulate the classification from unnecessary failure which generates a similarity measure to be used in resolving ambiguous cases. Many such metrics exist, such as the Levenshtein edit distance. The N-gram distance proves to be a fast method that performs well in the types of variations present in MetOc Web service descriptions. The IWB will then both index the service with a recording of the similarity value and utilize a GUI to allow expert user guidance in the disambiguation as illustrated in Figure 9. Specifically, terms from a Web service that were not exact matches for concepts in the ontology are evaluated as partial matches. The list of possible partial matches is returned to the IWB and a disambiguation window is then displayed on the IWB server monitor. This allows the user in charge of maintaining the IWB server to select which concept to index the Web service under. In order to assist the user in deciding which concept fits the Web service in question, links

to the Web service and WSDL are provided. If it is determined that the service is not a MetOc Web service, the user may click Not MetOc, and the service will not be indexed.

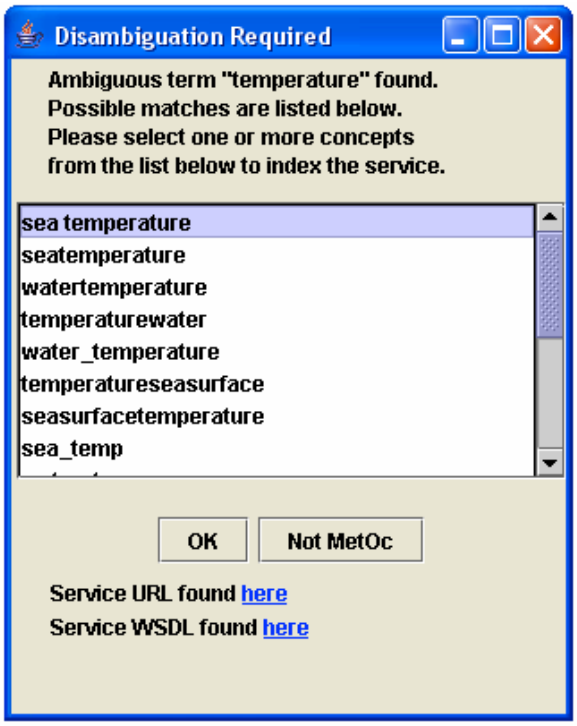


Figure 9. Sample Disambiguation Window for Term "temperature"

IWB Mediation of OGC Web Coverage Services

The Open Geospatial Consortium (OGC) Web Coverage Services (WCS) also provides METOC data description and we have also extended the capability of the IWB to integrate data from WCS sites. WCS supports retrieval of geospatial data as "coverages"—that is, geospatial information representing space-varying phenomena. WCS structurally differs from World Wide Web Consortium (WC3) Web Services standards (e.g., WSDL) but does utilize formal XML structures to provide three operations: GetCapabilities, DescribeCoverage and GetCoverage. We have found these three sufficient to integrate data in this format

A demonstration of IWB for mediation OGC data was conducted at the NATO Underwater Research Center (NURC)) at La Spezia Italy. This demonstration was in coordination with NURC's Turkish Straits Survey (TSS) exercise. End-to-end data brokering from the Turkish Straits Survey (TSS) to data consumers at Naval Research Lab (NRL) was provided by effectively integrating a NURC WCS into the IWB, including index and data retrieval. Figure 10 illustrates the environment for the demonstration. In the figure TSS data from RV Alliance (remote sensed, in-situ, model output) is delivered to NURC Data Fusion group. Data is then loaded into NURC

OpenGeoserver, published as OGC layers/KML. IWB indexes these layers and can answer JMBL queries with WCS metadata

It should be noted that future OGC plans are to provide a Web service capability for WCS similar to WC3 standards, which would facilitate use of IWB.

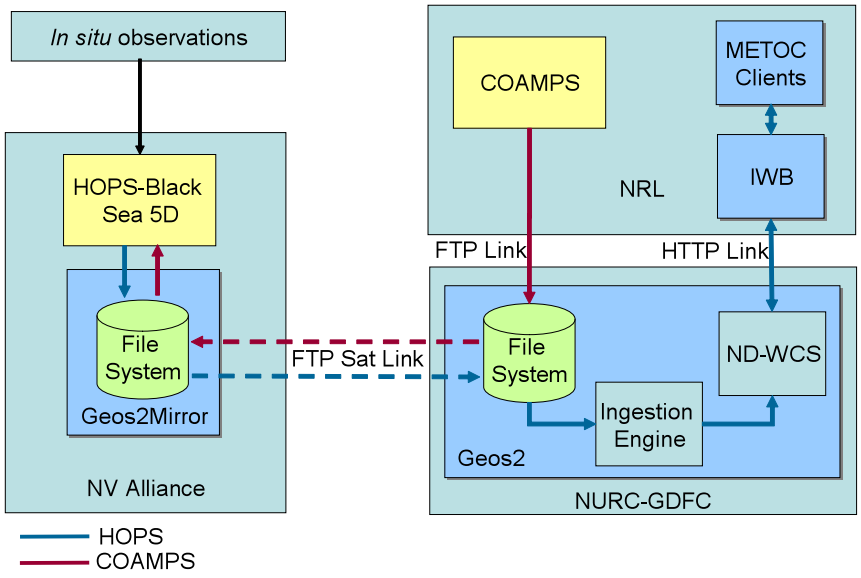


Figure 10. Turkish Straits Survey Data Flow

Development of MetOc Ontology for IWB

As we have discussed, development of complex ontologies are difficult and time consuming. We describe in this section the data sources and data descriptions we have used in the development of a MetOc ontology and give an example of such an ontology. Finally we introduce the possible use of a classifier to complement the discovery process.

MetOc Data and Sources

In this section we describe some relevant MetOc data and its available sources. This overview describes a diverse mix of data that will be seen to present a number of troublesome issues for ontology development.

For the development of the IWB, we focus on forecasts and observations of ocean and atmospheric parameters. Parameters of interest include measures of phenomena such as wave height, wave period, sea temperature, air temperature, pressure, and so forth. This forecasted data is generally derived from numerical models, which predict the measurement of conditions either at specific locations or over broad areas. When area data is produced, it may be a uniform grid in either two or three dimensions. Additionally, the forecast may include such data for multiple time-steps, showing environmental change over time. Representative of the nature of this output is the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) data. COAMPS provides short-term forecasts (usually up to 72 hours) for data such as air temperature and pressure, among others. Temporal granularity may be one hour increments or higher. Observation data represents the measurement of some condition at a specific location and time. It may include such parameters as wave height or wind speed, and it may be sourced from a variety of sensors, such as those that may be found on a buoy or weather balloon.

The type of data made available from the underlying forecast or observation can vary depending on the need of the user. Some of these include numeric output, narrative reports and graphic image products. Unlike numeric output, narratives might be used to provide a readable summary of a storm track. Image products may show graphics of symbolized features or color-coded data measurements. These may, for example, depict weather fronts or temperature variations on the ocean surface.

MetOc data is prepared and distributed by a number of government agencies and other sources. These include the Naval Oceanographic Office for ocean data and Fleet Numerical Meteorology and Oceanographic Center and the Air Force Weather Agency for atmospheric data. The National Oceanic and Atmospheric Administration (NOAA) and the National Data Buoy Center also provide such data for the atmosphere and ocean. NOAA, in particular, provides a wide range of data including weather information, ocean data on reefs, tides, currents, and so forth, real-time satellite imagery and data on large-scale climate conditions such as El Nino and global warning. The Argus Program includes seven stations around the world. These provide time exposure imagery that reveal sand bar movements. At some locations sensors record changing waves, winds, tides and currents on approximately an hourly basis. Many research facilities are also sources of oceanographic and atmospheric data. Some of these include Antarctic Cooperative Research Centre, the Center for Ocean Land Atmosphere Studies, Columbia

University/LDEO - International Research Institute, the Integrated Global Ocean Services System Products, the National Center for Atmospheric Research, among others.

Lack of naming convention uniformity among models and data providers is a significant issue when dealing with data retrieval in a distributed system. For example, the parameter name "temperature" may be used by different data producers to describe two very different temperatures—sea temperature and air temperature. The meaning may be known by the nature of the data provider, from the data itself or from associated documentation. One data provider, for example, may be known to produce ocean data, supporting the conclusion that the data is sea temperature. In other cases, units of measure in the data set may be those customarily used to describe isobar levels, supporting the conclusion that the data is air temperature.

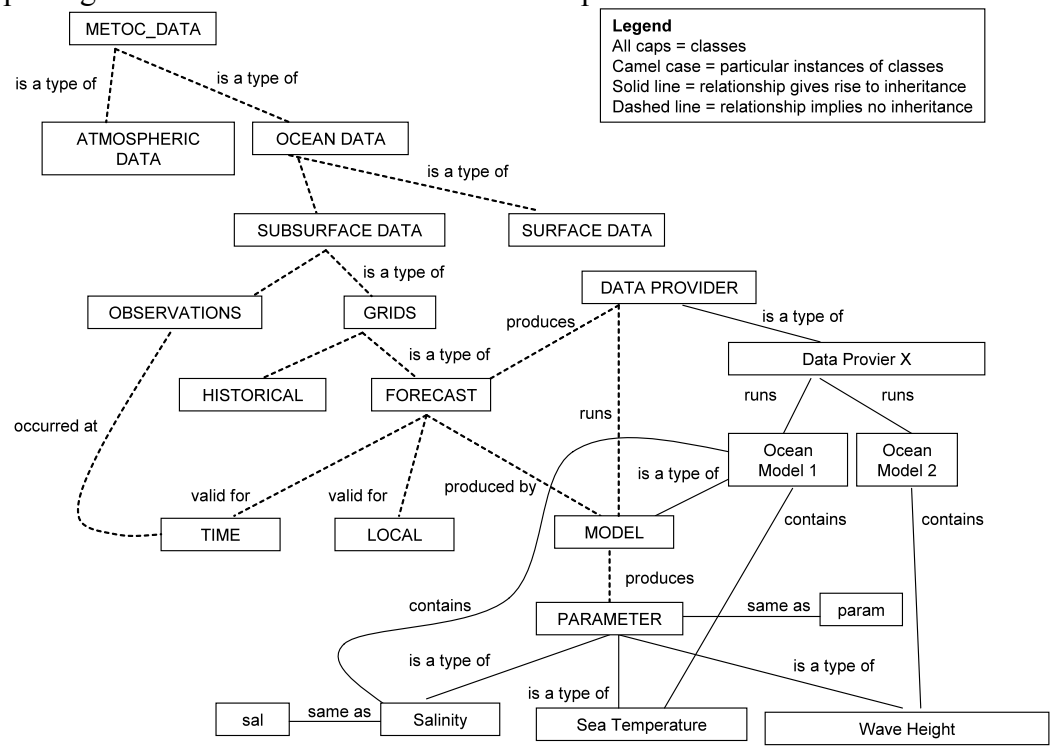


Figure 11. Sample of part of MetOc Ontology

Discussion of the MetOc Ontology

The sample ontology in Figure 11 captures a portion of both MetOc domain terminology and potential data source terminology. Some of the MetOc terms are quite general such as *surface* and *subsurface* and some very specific such as *salinity*. Other terms provide the descriptions of the data formats such as *grids* or *observations*. For the Web service identification it may be also useful or necessary to have data providers described along with the forecast models. This is just a simple illustration of the complexity involved in the development of such an ontology.

Because of this complexity in the development of domain ontologies, we have evaluated the use of case-based classification for some IWB tasks. Specifically this is intended as a complement to ontologies to support the automated discovery process for meteorological and oceanographic Web Services in our application. Case-based reasoning (CBR) is a problem solving methodology that retrieves and reuses decisions from stored cases to solve new problems (Kolodner, 1992), and case-based classification focuses on applying CBR to supervised classification tasks.

CLASSIFIER APPROACH

Identifying whether a given Web service supplies data for a particular domain can be framed as a classification or categorization task, which involves assigning one or more predefined labels to an unlabelled object. Thus, the Web Services identification task for the MetOc domain will involve assigning the label "MetOc" or "Non-MetOc" to a given Web service. In this section, we describe our approach, which uses nearest-neighbor or case-based classification. Finally, we evaluate our methodology using meteorological Web Services.

Overview of Classification Approaches

Our goal is to automatically build classifiers from example data, often termed *supervised learning*. To formally describe a supervised classifier learning approach, we first present the relevant notation. The example data required for classifier learning should be in a tabular format, where each row in the table is an object *o* and each column in table is an attribute *a* (see Table 1). Let ***O*** represent set of objects in the table and ***A*** represent the set of attributes (i.e., columns) in the table. Each cell in the table is the value v_{ij} of the attribute for $\boldsymbol{a}_j$ for a particular object o_i. We partition the attributes into two types: (1) *Conditional attributes* denoted by ***C,*** *which* are the object characteristics that provide information for classification and (2) *Decision attribute(s)* ***D,*** which are attributes whose values indicate the category that applies to an object.

Table 1. Example Web Services data for classifier learning

		Attributes -***A***					
		Conditional Attributes -***C***					Decision Attribute ***D***
		c_1 (zipcode)	c_2 (temperature)	c_3 (water)	c_4 (price)	c_5 (get)	*d*
O	o_1	3	2	1	0	1	Metoc
	o_2	1	0	0	2	3	Non-Metoc
	o_3	1	1	0	2	3	Non-Metoc
	o_4	2	1	4	1	4	Metoc

Learning a classifier implies finding the function h that maps objects in $\boldsymbol{O}$ to decisions in $\boldsymbol{D}$, that is, h: $\boldsymbol{O} \rightarrow \boldsymbol{D}$. The methods for estimating or learning h depend on the family of functions under consideration. For example, linear and non-linear regression techniques, neural networks, decision tree learning, support vector machines, and nearest neighbor techniques are some of the methods used for building classifiers (Tan, Steinbach, & Kumar, 2006). Different classifiers have various strengths and weaknesses depending on the nature and the amount of example data. Typically, most classifiers are hard to develop when the data has a large number of attributes (in thousands), missing values, and only a few example objects (< 100 per class). Many applications have such characteristics, especially those that deal with attributes that are textual in nature. Email classification and text categorization have attributes that run into 1000s (Gupta, Moore, Aha, & Pal, 2005). For such applications, case-based or nearest-neighbor classification approaches have been shown to be effective.

Case-Based Classification

Case-based classification proceeds as follows. To classify a new object, the classification decision from previously classified objects is reused. Objects that have characteristics similar to the new object are called cases. Each object in the Table 1 is a case, and the list of objects in the table constitutes the casebase. To assess the similarity of one case with another, the classifier uses a similarity metric or a matching function such as the Euclidean distance metric used as a similarity function. The cases that are most similar to the unclassified object are called the nearest neighbors. The decisions from the k nearest neighbors from the case base are used in assigning the class label to a new object. Training the classifier typically implies estimating the weights or parameters applicable to the similarity metric.

Web service classification in the MetOc application entails assigning one of the following two labels, "MetOc" or "non-MetOc", to a Web service in question. The input to the classifier is a Web service schema described using the WSDL and the output is an associated label. Prior to using the classifier, it must be trained on example cases as follows (see Figure 12):

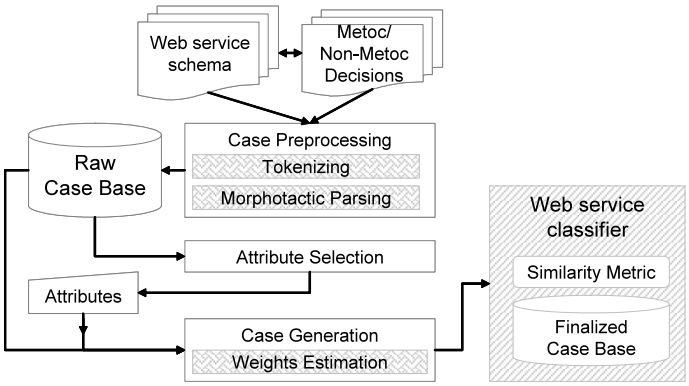


Figure 12. Web Service Classifier Training Process

1. *Case preprocessing*:

For classification, each WSDL and its associated schema must be converted into a case with attributes and values. We treat all the element contents in a schema as a source of attributes. For example, an element in a schema may contain the text "waterTemperature." Alternatively, to reduce sparseness of cases, the element can be broken down by the tokenization process into constituent terms. So "waterTemperature" is broken into "water" and "temperature". Subsequently, the morphotactic parsing process further reduces words into their baseforms (Gupta & Aha, 2005). The word "producer" is reduced to its baseform "produce". This then has transformed the Web service schema to a bag of unique baseforms. Each baseform is a potential case attribute and the frequency of its occurrence in a particular schema is the value corresponding to it. This is stored as a raw case in a preliminary case base. For each case, the decision of whether it is MetOc or non-MetOc is added as the decision attribute.

2. *Attribute selection*:
   With potentially hundreds of example Web services for classifier training, we expect to generate thousands of attributes. This is a serious computational challenge and can also adversely affect classification performance by introducing noisy and irrelevant attributes. For example, the attribute "http" may appear in all cases and provide no useful information to discriminate MetOc from non-MetOc Web services. To counter this problem, we perform attribute selection, where a metric is used to select a subset of attributes with a potential to improve classification performance. Attribute selection metrics such as mutual information, information gain, document frequency (Yang & Pederson, 1997), and rough set methods can be used (Gupta et al., 2005) to select attributes. We apply the information gain metric to select attributes in the Web Services Classifier.

3. *Case Generation*:
   After the attributes have been selected, each case must be indexed with the selected attributes and their corresponding weights must be computed. We use the information gain metric to calculate the weights applicable to the attributes. This outputs a classifier that includes the finalized cases and the similarity metric.

After training is complete, the classification of a new Web service proceeds as follows. First, case preprocessing and case generation processes are used to convert the Web service schema into a case. This case is matched with the cases in the case base using the learned similarity metric and the k-nearest neighbors are retrieved. The decision from the retrieved cases is then applied to the new case as follows. Each nearest neighbor votes on the decisions based on its classification. Each vote is weighted by the similarity of the voting neighbor. The classification label with the most votes is assigned as the decision to the new case. If the decision assigned to the new case is the same as expected, then it is counted as a correct classification or else a wrong classification. The classifier performance is measured by the percentage of cases classified correctly.

Evaluation

We have evaluated the Web Services Classifier with in-lab testing and operationally. For in-lab testing, a set of 64 Web service schemas was obtained from the registries on the Web. Our meteorological subject matter expert classified 26 of these schemas as MetOc relevant. We followed the leave-one-out method of performance evaluation for the classifier. In the leave one out method, one case is taken out of the set as a test case and the remaining cases are used to train the classifier. The classification accuracy for the test case is recorded using the trained classifier. This process of training and classification is repeated for each case in the set to evaluate classification accuracy.

Using the above data and leave one out testing, we obtained a maximum classification accuracy for the Web Services Classifier of 93.75%. The number of nearest neighbors used for this classification was 5, and the number of attributes used in the process was 523, which reflected a reduction from 1790 total possible attributes. The optimal parameters were obtained by a genetic algorithm that used the classification accuracy as its fitness function.

In addition to this in-lab testing, we have examined the operational performance of the classifier against two registries, xmethods and Webservicex. These results are shown in Table 2. In this experiment, the number of total WSDLs in xmethods were 384 and in Webservicex 66. The results showed that the classifier correctly classified 97.40% of the Web services in xmethods and 97.01% of the Web services in the Webservicex registry as being MetOc or non-MetOc. A significant finding was that there were no false negatives for the xmethods registry and only two for the Webservicex registry. This indicates that no substantial sources of MetOc data were overlooked by the classifier.

Table 2. Classifier Operational Performance

Registry Used	Classified non-MetOc	Classified MetOc	Number False Positives	Number False Negatives	Accuracy	Recall
xmethods	368	16	10	0	97.40%	100.00%
Webservicex	64	2	0	2	97.01%	50.00%

DISCUSSION AND CONCLUSIONS

There has been considerable research on ontologies to help resolve difficulties of sharing knowledge among various domains of interest. In some uses of ontologies by Web services, data providers are assumed to deploy an ontological description of their Web service to support automated discovery and integration by interested client applications (Paolucci, Saudry, Srinivasan, & Sycara, 2004). The IWB approach of using a dynamic knowledge base does not require such ontological descriptions.

There has also been some research on Web Services Classification as a means of automating or semi-automating the annotation of Web Services with semantic meaning. That work has had as its focus the automatic generation of Web Services ontologies such as OWL-S (Heß & Kushmerick, 2003, 2004). In contrast to the use of OWL-S, we have investigated the use of classifiers for Web Services discovery.

In this paper we have presented the general design principles of an Integrated Web Services Brokering System (IWB). The IWB embodies an end-to-end system that

performs automated discovery, automated mediation and automated request/response transformation and is in the specific context of the domain of meteorological and oceanographic Web Services. While we utilize domain specific ontologies, we have also examined the feasibility of case-based classification to support automated Web Services discovery.

ACKNOWLEGMENT

The authors would like to thank the Naval Research Laboratory's Base Program, Program Element No. 0602435N for sponsoring this research.

REFERENCES

Cerami, E. (2002). *Web services essentials*. Sebastopol, CA: O'Reilly and Associates.

Chao, K., Younas, M., Lo, C., & Tan, T. (2005, March 28-30). Fuzzy matchmaking for Web services. In *Proceedings of the IEEE 19th International Conference on Advanced Information Networking and Applications,* Taiwan (pp. Vol. 2, pp. 721-726). Washington, DC: IEEE Computer Society.

Corcho, O., Fernandez-Lopez, M., & Gomez-Perez, A. (2007). Ontological engineering: What are ontolgoies and how can we build them? In J. Cardoso (Ed.), *Semantic Web services* (pp. 44-70). Hershey, PA: IGI Global.

Cristani, M., & Cuel, R. (2004). A survey on ontology creation methodologies. *International Journal on Semantic Web and Information Systems, 1*(2), 49-69.

Dick, K. (2000). *XML: A manager's guide.* Reading, MA :Addison Wesley.

Fenza, G., Loia, V., & Senatore, S. (2007, July 23-26). Improving fuzzy service matchmaking through concept matching discovery. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, London (pp. 1-6). Washington, DC: IEEE Computer Society.

Fenza, G., Loia, V., & Senatore, S. (2008). A hybrid approach to Semantic Web service matching. *International Journal of Approximate Reasoning, 48*(3), 808-828.

Gupta, K., Moore, P.G, Aha, D., & Pal, S. (2005, December 18-22). Rough-set feature selection methods for case-based categorization of text documents. In S. K. Pal, S. Bandyopadhyay, & S. Biswas (Eds.), *Proceedings of the 1st International Conference on Pattern Recognition and Machine Intelligence*, Kolkata, India (LNCS 3776, pp. 792-798).

Gupta, K. M., & Aha, D. W. (2004, December 19-22). RuMoP: A morphotactic parser. In *Proceedings of the International Conference on Natural Language Processing,* Hyderabad, India (pp. 280-284). Springer.

Heß, A., & Kushmerick, N. (2003, October 20-23). Learning to attach semantic metadata to Web services. In *Proceedings of the 2nd International Semantic Web Conference*, Sanibel Island, FL (pp. 258-273). AAAI Press.

Heß, A., & Kushmerick, N. (2004, March 22-24). Machine learning for annotating Semantic Web services. In *Proceedings of the AAAI Spring Symposium on Semantic Web Services,* Palo Alto, CA (pp. 341-346). AAAI Press.

Holsapple,C., & Joshi, K. (2002). A collaborative approach to ontology design. *Communications of the ACM, 45*(2), 42-47.

JMBL. (2009). *Joint METOC public data administration Web site*. Retrieved from http://www.cffc.navy.mil/metoc

Knublauch, H., Fergerson, F., Noy N., & Musen, M. (2004, November 7-11). The Protege OWL plugin: An open development environment for Semantic Web applications. In *Proceedings of the 3rd International Conference on the Semantic Web (ISWC-2004),* Hiroshima, Japan (pp. 342-351).

Kolodner, J. (1992). An introduction to case-based reasoning. *Artificial Intelligence Review, 6*, 3-24.

Kim, H. (2002). Predicting how ontologies for the Semantic Web will evolve. *Communications of the ACM, 45*(2), 48-54

Lacy, L. (2005). *Owl: Representing information using the Web ontology language*. London: Trafford Publishing.

Ladner, R., & Petry, F. (2005). *Net-centric approaches to intelligence and national security*. Dordrecht, the Netherlands: Kluwer Academic Press.

Obrst, L. (2003, November). Ontologies for semantically interoperable systems. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM'03),* New Orleans, LA (pp. 366-369). ACM Publishing.

Paolucci, M., Soudry, J., Srinivasan, N., & Sycara, K. (2004). A broker for OWL-S Web services. In *Proceedings of the AAAI Spring Symposium on Semantic Web Services,* Palo Alto, CA (pp. 562-567). AAAI Press.

Papazoglou, M. (2007). *Web services: Principles and technology*. Upper Saddle River, NJ: Prentice Hall.

Powers, S. (2003). *Practical RDF*. Sebastopol, CA: O'Reilly Media.

Rama, A. (2007). Semantic Web services. In J. Cardoso (Ed.), *Semantic Web services* (pp. 191-216). Hershey PA: IGI Global.

Stojanovic, L., Schneider, J., Maedche, A., Libischer, S., Studer, R., Lumpp, T., et al. (2004). The role of ontologies in autonomic computing systems. *IBM Systems Journal, 43*(3), 598-616.

Tan, P., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining*. Boston: Pearson.

van der Vet, P., & Mars, N. (1998). Bottom-up construction of ontologies. *IEEE Transactions on Knowledge and Data Engineering, 16*(4), 513-526.

W3C Member Submission. (2004). *OWL-S: Semantic markup for Web services*. Retrieved from http://www.w3.org/Submission/OWL-S/

Yang, Y., & Pederson, J. (1997, July 8-12). A comparative study of feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning,* Nashville, TN (pp. 412-420). Morgan Kaufmann.